

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

DWES

Formularios en PHP

2º DAW

Uso de formularios desde PHP

- Los formularios nos permiten recoger datos de una página web y enviarlos a una dirección URL que será una página PHP que los recogerá y los procesará.
- Dentro de los controles del formulario debemos destacar dos atributos:
 - ✓ `method`: Por defecto toma el método GET, pero el más recomendable es el método POST que envía los datos en el cuerpo del mensaje en lugar de en la URL, como ocurre con el método GET.
 - ✓ `action`: indica la URL del fichero que recogerá los datos.
- Para que desde PHP podamos acceder a los datos que se envían por el formulario, **todos los elementos que queremos que se envíen tienen que tener definido el atributo `name` con un valor.**
- Cuando un elemento del formulario no se rellena o no selecciona, la variable no se crea, y por tanto desde la página PHP se puede comprobar con la función `isset()`
- Los elementos del formulario se reciben en unas variables superglobales `$_GET`, `$_POST` o `$_REQUEST` que contendrán los elementos del formulario que se han recibido en un array asociativo.

Uso de formularios desde PHP

La información se puede enviar a PHP desde el cliente mediante dos métodos:

➤ GET

- La información se envía como parte de la URL en parejas “variable=valor”.
- Es el método por defecto de envío de información tanto en los enlaces (siempre) como en los formularios HTML.
- Si la información a enviar es sensible no es recomendable.

➤ POST

- La información se pasa en el cuerpo de la petición, con lo que ya no es visible en la URL.
- Es el más recomendable cuando usamos formularios HTML.

Uso de formularios desde PHP

GET

Por ejemplo tenemos la siguiente URL:

<https://miservidor.com/prueba.php?id=2300&precio=24>

- **prueba.php** indica la página a la que se redirige el programa
- El símbolo **?** marca el comienzo de los parámetros que se envían, en este hay dos parámetros, el primero con nombre **id** y valor **2300** y el segundo con nombre **precio** y valor **24**.
- Los pares parámetro=valor se separan en la URL con el símbolo **&**
- **Recuerda que esta información puede verse en la casilla de dirección del navegador.**

Uso de formularios desde PHP

Recoger los datos pasados por GET desde PHP

- La variable superglobal `$_GET` es un array asociativo que contiene todas las variables y sus valores que se han pasado al servidor utilizando el método GET.
- Se debe comprobar si existe o no (mediante la función `isset()`) lo que indicará si han llegado valores o no.
- Para obtener el valor de una variable solo hay que escribir `$_GET['nombre variable'];`
- Por ejemplo: `$precio = $_GET['precio'];`

Ejemplo de paso de información por un enlace

```
<?php
    echo "<a href= 'pagina2.php?variable1 = Hola&variable2 = 1234'> texto del
enlace</a>";

?>
```

```
<?php
    echo $_GET["variable1"];

?>
```

Uso de formularios desde PHP

Recoger los datos pasados por POST desde PHP

- La variable superglobal **`$_POST`** es un array asociativo que contiene todos las variables y sus valores que se han pasado al servidor utilizando el método POST.
- Se debe comprobar si existe o no (mediante la función `isset()`) lo que indicará si han llegado valores o no.
- Para obtener el valor de una variable solo hay que escribir `$_POST['nombre variable'];`
- Por ejemplo: **`$precio = $_POST['precio'];`**

Recoger los datos pasados por GET y POST desde PHP

- La variable superglobal **`$_REQUEST`** es un array asociativo que contiene todos las variables y sus valores que se han pasado al servidor utilizando ambos métodos POST y GET.
- Por ejemplo: **`$precio = $_REQUEST['precio'];`**

Uso de formularios desde PHP

Ejemplo: envío de datos de un formulario por método GET

- ✓ Tenemos un formulario HTML con dos campos de tipo input y un input de tipo submit que corresponde con el botón de envío de formulario.
- ✓ Todos los campos susceptibles de ser enviados (incluso el botón de envío) deben tener valor en el atributo **name**.
- ✓ El método de envío es GET y la página a la que se envían los datos del formulario es "formulario_1.php"

```
<!DOCTYPE html>
<html>
  <head>
    <title>Formularios</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form name="frmPrueba" method="get" action="formulario_1.php" >
      <input type="text" name="txtPrimero" id="txt1" />
      <input type="text" name="txtSegundo" id="txt2"
        value="texto predefinido" />
      <input type="submit" name="btnEnviar" value="Enviar Formulario" />
    </form>
  </body>
</html>
```

Observa que el botón de envío del formulario también tiene definido un valor en el atributo name. Esto nos permitirá comprobar desde otro archivo si se ha pulsado (si existe, es que se ha pulsado, el elemento se crea. Si no existe, no se ha pulsado, el elemento no se crea)

Uso de formularios desde PHP

Ejemplo: envío de datos de un formulario por método GET

- ✓ Una vez enviado el formulario se procesará por la página *formulario_1.php*:

```
<?php
/* Comprobar si el formulario fue enviado (si se pulsó el botón enviar)
 */
if(isset($_GET['btnEnviar']))
{
    echo "<p>Contenido del array asociativo con la información del "
    . "formulario:</p>";
    print_r( $_GET );
    echo "<p/>";
    echo "Primer texto: [".$_GET['txtPrimero']."]<br/>";
    echo "Segundo texto: [".$_GET['txtSegundo']."]<br/>";
    echo "<hr/>";
}
else{
    echo "No hay datos del formulario <br/>";
}
?>
```

En esta página (formulario_1.php) recuperamos los datos del formulario que se han enviado. Como se enviaron utilizando el método GET debemos recuperarlos usando `$_GET` o `$_REQUEST`. La clave del array asociativo de cada elemento es el valor del atributo name que se puso en el formulario.

Uso de formularios desde PHP

Ejemplo: envío de datos de un formulario por método POST

- ✓ Partimos del ejemplo anterior (un formulario HTML con dos campos de tipo input y un input de tipo submit que corresponde con el botón de envío de formulario).
- ✓ La diferencia está en el método de envío, ahora usamos POST en lugar de GET.

• Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form name="frmPrueba" method="post" action="formulario_1.php" >
      <input type="text" name="txtPrimero" id="txt1" />
      <input type="text" name="txtSegundo" id="txt2" value="texto predefinido" />
      <input type="submit" name="btnEnviar" value="Enviar Formulario" />
    </form>
  </body>
</html>
```

Tenemos un formulario con dos campos de tipo input y un input de tipo submit que corresponde con el botón de envío de formulario.

El método de envío es POST y la página a la que se envían los datos del formulario es "formulario_1.php"

Uso de formularios desde PHP

Ejemplo: envío de datos de un formulario por método POST

- ✓ Una vez enviado el formulario se procesará por la página *formulario_1.php*:

```
<?php
/* Comprobar si el formulario fue enviado (si se pulsó el botón enviar)
 */
if(isset($_POST['btnEnviar']))
{
    echo "<p>Contenido del array asociativo con la información del "
    . "formulario:</p>";
    print_r( $_POST );
    echo "<p/>";
    echo "Primer texto: [".$_POST['txtPrimero']."]<br/>";
    echo "Segundo texto: [".$_POST['txtSegundo']."]<br/>";
    echo "<hr/>";
}
else{
    echo "No hay datos del formulario <br/>";
}
?>
```

En la página *formulario_1.php* recuperamos los datos del formulario que se han enviado. Como se enviaron utilizando el método POST debemos recuperarlos usando `$_POST` o `$_REQUEST`. La clave del array asociativo de cada elemento es el valor del atributo `name` que se puso en el formulario.

Uso de formularios desde PHP

Cómo recibir del formulario los siguientes tipos de datos:

- ☐ Datos tipo radio
- ☐ Datos tipo checkbok
- ☐ Datos tipo select



Radio

- ✓ Los elementos de tipo radio **permiten seleccionar una y solo una opción** de entre varias opciones (son opciones excluyentes)
- ✓ Cada elemento de tipo radio del mismo grupo de opciones debe tener el mismo valor en el atributo name.
- ✓ Cuando se reciben los datos, solo llegará el valor elemento radio que fue seleccionado.



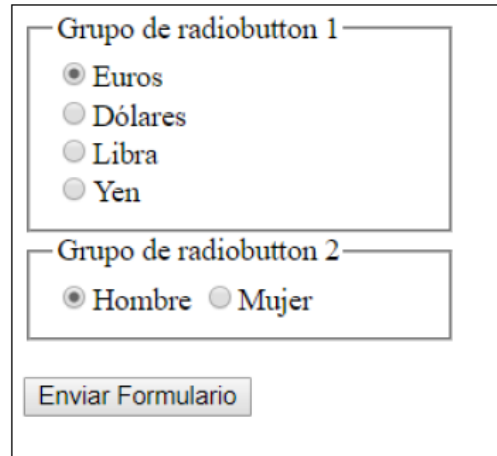
Checkbox

- ✓ Los elementos de tipo checkbox **permiten seleccionar varias opciones** de entre un grupo.
- ✓ En cada elemento checkbox el valor del atributo name será diferente.
- ✓ Cuando se reciben los datos, solo llegarán los valores de los elementos checkbox que fueron seleccionados.

Uso de formularios desde PHP

❑ Ejemplo 1 - Datos tipo radio

```
<form method="post" action="procesa.php" style="width:20%;">
  <fieldset>
    <legend>Grupo de radiobutton 1</legend>
    <input type="radio" name="moneda" value=1 checked/>Euros<br/>
    <input type="radio" name="moneda" value=2/>Dólares<br/>
    <input type="radio" name="moneda" value=3/>Libra<br/>
    <input type="radio" name="moneda" value=4/>Yen<br/>
  </fieldset>
  <fieldset>
    <legend>Grupo de radiobutton 2</legend>
    <input type="radio" name="sexo" value="hombre" checked/>Hombre
    <input type="radio" name="sexo" value="mujer" />Mujer
  </fieldset>
  <br/>
  <input type="submit" name="btnEnviar" value="Enviar Formulario" />
</form>
```



Todos los atributos “name” que pertenecen al mismo grupo de “radio”, tienen el mismo valor en el atributo name.

El contenido del atributo **value** es el valor del elemento con que llegará a la página el elemento seleccionado.

Uso de formularios desde PHP

❑ Ejemplo 1 - Datos tipo radio

```
<?php
/* comprobamos si existe el botón enviar del formulario, eso significa
 * que se ha pulsado el botón ya que existe.
 */
if(isset($_POST['btnEnviar'])){
    switch($_POST['moneda']){
        case 1:
            $moneda = "Euros";
            break;
        case 2:
            $moneda = "Dólares";
            break;
        case 3:
            $moneda = "Libras";
            break;
        case 4:
            $moneda = "Yenes";
            break;
    }
    echo "El valor que llega del primer grupo de elementos es <b>$moneda</b> <br/>";
    echo "El valor que llega del primer grupo de elementos es <b>".$_POST['sexo']."</b><br/>";
}
```

Código que recupera los datos del formulario enviados y salida por pantalla. Comprueba cuál es el valor (value) que ha llegado.

El valor que llega del primer grupo de elementos es **Dólares**
El valor que llega del segundo grupo de elementos es **hombre**

Uso de formularios desde PHP

❑ Ejemplo 2 - Datos tipo checkbox

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <form method="POST" action="procesa.php" >
      <input type="checkbox" name="futbol" value="11">Fútbol<br>
      <input type="checkbox" name="baloncesto" value="5">Baloncesto<br>
      <input type="checkbox" name="balonmano" value="7">Balonmano<br>
      <input type="checkbox" name="tenis" value="1 o 2">Tenis<br>
      <input type="submit" name="btnComprobar" value="Enviar" />
    </form>
  </body>
</html>
```

Observa que los atributos name de los elementos checkbox tienen un valor diferente.

El atributo value es el valor del elemento que se recibirá en la página que procesa los datos del formulario si dicho elemento se seleccionó.

Uso de formularios desde PHP

☐ Ejemplo 2 - Datos tipo checkbox

Observa que se accede a cada elemento de tipo checkbox mediante el array asociativo `$_POST`, siendo la clave de cada elemento el valor que se indicó en su atributo name.

```
if(isset($_POST['btnComprobar'])) {  
    if(isset($_POST['futbol']))  
        echo "El fútbol se juega con " . $_POST['futbol'] . " jugadores<br/>";  
    if(isset($_POST['baloncesto']))  
        echo "El baloncesto se juega con " . $_POST['baloncesto'] . " jugadores<br/>";  
    if(isset($_POST['balonmano']))  
        echo "El balonmano se juega con " . $_POST['balonmano'] . " jugadores<br/>";  
    if(isset($_POST['tenis']))  
        echo "El tenis se juega con " . $_POST['tenis'] . " jugadores<br/>";  
}
```


Uso de formularios desde PHP

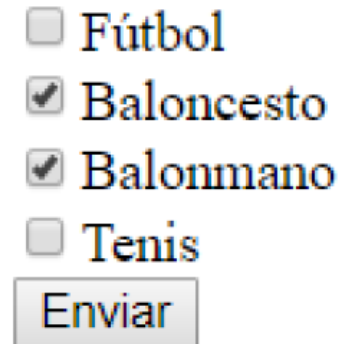
❑ Ejemplo 2 - Datos tipo checkbox

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <form method="POST" action="procesa.php" >
      <input type="checkbox" name="deportes[]" value="11">Fútbol<br>
      <input type="checkbox" name="deportes[]" value="5">Baloncesto<br>
      <input type="checkbox" name="deportes[]" value="7">Balonmano<br>
      <input type="checkbox" name="deportes[]" value="1 o 2">Tenis<br>
      <input type="submit" name="btnComprobar" value="Enviar" />
    </form>
  </body>
</html>
```

El método anterior puede ser muy pesado si tenemos muchos elementos de tipo checkbox en un mismo formulario.

Podemos agruparlos en un array poniendo en el atributo name de cada elemento el nombre de un array (el mismo) seguido de un par de corchetes vacíos (recuerda que esa es la forma de añadir elementos al final de un array).

De esta forma, al enviar el formulario recibiremos un array con tantos elementos como elementos checkbox fueron **SELECCIONADOS**, que se podrá recorrer mediante un bucle.



Visual representation of the HTML form:

- ☐ Fútbol
- ☒ Baloncesto
- ☒ Balonmano
- ☐ Tenis
-

Uso de formularios desde PHP

❑ Ejemplo 2 - Datos tipo checkbox

El deporte con indice 0 se juega con 5 jugadores
El deporte con indice 1 se juega con 7 jugadores

```
<?php
    if(isset($_POST['btnComprobar'])) {
        foreach($_POST['deportes'] as $clave=>$valor){
            echo "El deporte con indice $clave se juega con $valor jugadores<br/>";
        }
    }
}
```

Al no poner nombre de clave en cada elemento del array en el formulario, se crea un array indexado con elementos **SELECCIONADOS** consecutivos, es decir el elemento con índice 0 será el primer **elemento checkbox seleccionado**, el índice 1 corresponde con el **segundo elemento checkbox seleccionado**, etc.

Uso de formularios desde PHP

❑ Datos tipo select

- ✓ Los elementos de tipo select **permiten seleccionar de una lista desplegable** una opción.
- ✓ El elemento con etiqueta <select> es al que tiene que tener valor en el atributo name.
- ✓ Cuando se reciben los datos, llegará el valor (value) del elemento que fue seleccionado.

```
<!--Formulario -->
<select name="selCombo">
  <option value="valor1">1-1000</option>
  <option value="valor2">1001-3000</option>
  <option value="valor3"> >3000</option>
</select>

.....
// código PHP
if ($_POST["selCombo"]=="valor3"){ .....
```



Uso de formularios desde PHP

❑ Ejercicio propuesto: Conversor de moneda

Realiza una aplicación web que realice conversión de una cantidad de euros que se introducirá por teclado al tipo de moneda que se seleccione.

- El formulario debe permitir elegir un tipo de moneda de entre bitcoin, dólar americano, libra esterlina y yen japonés utilizando un tipo select.
- El formulario se enviará a otro archivo PHP que será el que haga el cálculo y muestre el resultado por pantalla.
- La tasa de cambio a aplicar es:
 - ✓ 1€= 0,00012 ₿ (Bitcoin)
 - ✓ 1€= 1,12 \$ (dólar americano)
 - ✓ 1€= 0,86 £ (libra esterlina)
 - ✓ 1€= 120,82¥ (yen japonés)

Ampliación: Cambia el código del ejercicio de conversión de moneda para que se pueda elegir más de un tipo de moneda a convertir (checkbox). El programa mostrará el cambio en todos los tipos de moneda seleccionados.